

**NASA TECHNICAL  
MEMORANDUM**

NASA TM X- 62,324

NASA TM X- 62,324

(NASA-TM-X-62324) INFLUENCE MATRIX  
PROGRAM FOR AERODYNAMIC LIFTING SURFACE  
THEORY (NASA) 62 p HC CSCL 01A

N74-16701

Unclas

G3/01 29935

**INFLUENCE MATRIX PROGRAM FOR AERODYNAMIC  
LIFTING SURFACE THEORY**

Richard T. Medan and K. Susan Ray

Ames Research Center  
Moffett Field, California 94035

and

Computer Sciences Corporation  
Mountain View, California 94043

December 1973

REPRODUCED BY  
**NATIONAL TECHNICAL  
INFORMATION SERVICE**  
U. S. DEPARTMENT OF COMMERCE  
SPRINGFIELD, VA. 22161

62

## TABLE OF CONTENTS

	ABSTRACT	iii
1	INTRODUCTION	1
2	PROGRAM DESCRIPTION	2
3	INSTRUCTIONS	4
	3.1 INITIAL SETUP FOR AMES' TSS SYSTEM	4
	3.2 TSS CONVERSATIONAL MODE	4
	3.3 TSS BATCH MODE	5
4	DESCRIPTION OF COMMANDS	6
	CC1	7
	CONTINUE	7
	CWTYPE	7
	DELTA0	7
	EPS	8
	IWCCP	8
	IWRITE	8
	IWSCP	8
	JJ	8
	KK	8
	LMAX	9
	LMIN	9
	MACH	9
	MM	9
	MODES	10
	MREF	10
	NINT	10
	NMAX	10
	NOHEAD	11
	NSTORE	11
	PP	11
	START	11
	STOP	11
	SUPPRESS	12
	SWTYPE	12
	TSS	12
	WING	12

5	SAMPLE TERMINAL SESSION	13
6	INPUT FILES	16
	6.1 GEOMETRY FILE	16
	6.2 IDENTIFICATION NUMBER FILE	16
7	INFLUENCE MATRIX (AIM) FILE	17
8	REFERENCES	20
	FIGURES	21
	APPENDIX 1--OUTPUT FROM SAMPLE SESSION	24
	APPENDIX 2--PROGRAM LISTING	33

INFLUENCE MATRIX PROGRAM FOR  
AERODYNAMIC LIFTING SURFACE THEORY

Richard T. Medan  
Ames Research Center  
and  
K. Susan Ray  
Computer Sciences Corporation

ABSTRACT

This document is a description of and users manual for a USA FORTRAN IV computer program which computes an aerodynamic influence matrix and which is one of several computer programs used to analyze lifting, thin wings in steady, subsonic flow according to a kernel function method lifting surface theory. The most significant features of the program are that it can treat unsymmetrical wings, control points can be placed on the leading and/or trailing edges, and a stable, efficient algorithm is used to compute the influence matrix.

INFLUENCE MATRIX PROGRAM FOR  
AERODYNAMIC LIFTING SURFACE THEORY

Richard T. Medan  
Ames Research Center  
and  
K. Susan Ray  
Computer Sciences Corporation

1 INTRODUCTION

This document is a description of and users manual for a USA FORTRAN IV computer program which computes an aerodynamic influence matrix and which is one of several computer programs used to analyze lifting, thin wings in steady, subsonic flow according to a kernel function method lifting surface theory. The most significant features of the program are that it can treat unsymmetrical wings, control points can be placed on the leading and/or trailing edges, and a stable, efficient algorithm is used to compute the influence matrix.

Questions concerning either this document or the computer program or the associated computer programs should be directed to:

R. T. Medan  
Mail Stop 247-1  
Ames Research Center  
Moffett Field,  
Ca. 94035

## Influence Matrix Program

### 2 PROGRAM DESCRIPTION

This program implements the algorithms discussed in ref. 1 for determining the aerodynamic influence matrix of a thin, lifting, planar wing in linearized, steady, subsonic flow. (i.e., the downwash modes induced by the pressure modes). The program utilizes the new technique (discussed in ref. 1) which requires fewer quadrature points yet still calculates the influence functions accurately enough to guarantee convergence with an increasing number of spanwise quadrature points.

The types of wings which may be treated by the program include asymmetric ones and ones with curved edges. Also control points on the wing leading and/or trailing edges may be handled. The wing geometry and spanwise integration points are input to the program from a disk file or tape (see section 5.1) created by the geometry program (ref. 2). The influence matrix which this program calculates is written on a disk file or tape (AIM file, see section 5.2 for a complete description). This AIM file is then used by the equation solving program (ref. 3), which determines the coefficients in the expansion for the lifting pressure coefficient,  $\Delta C_p$ . Also the AIM file may be used by the boundary condition program (ref. 4) and the planform plotting program (ref. 5) since the wing control points may be changed by this influence matrix program.

This program was designed to operate under a command format. After some initial input is entered, the program prints a + sign (in the conversational mode). At this point the user enters a command. Then the program performs an action associated with the command. This action may consist of reading a variable or variables, reading in the geometry of a new wing, or beginning the computation of the AIM file for the current wing.

This command format approach has been found to be extremely flexible and to make the program very easy to use. For example, it allows the user to enter data in any order that he wishes and allows a conversational user to correct mistakes easily since each command may be given as many times as the user wishes. A complete description of the available commands is given in

## Influence Matrix Program

### section 4.

Another particularly useful feature of this program is that, with a single program run, the user can create more than one AIM file. These files can be AIM files for different wings and/or AIM files for the same wing. AIM files for the same wing can differ from one another because of changes in the Mach number or else different choices for the number of spanwise integration points or other integration parameters or else because of a different set of control points. The ability of the program to change the chordwise control points arbitrarily and the spanwise control points to a limited extent from those (default) control points stored on the geometry file has been found to be another very useful feature.

## Influence Matrix Program

### 3 USERS INSTRUCTIONS

#### 3.1 INITIAL SETUP AMES' TSS SYSTEM

For either batch or conversational processing the following TSS commands must be given. These commands are required once and only once for each user ID. The first three commands create the identification number file named IDFILE. This file contains four zeroes in binary form.

```
SHARE MEDAN,FSARTM,INIDFILE
CDS MEDAN,IDFILE
DELETE MEDAN
SHARE MEDAN,FSARTM,LSPROG.V1
```

#### 3.2 CONVERSATIONAL USE ON AMES' TSS SYSTEM

All integer data should be entered in a 1615 format and all floating point data in 8F10.0 format.

USER: After logging on enter the following:

```
AMES USYSLIB
JOBLIBS SYSULIB
JBLB MEDAN
```

It is not necessary to issue DDEFS for the geometry, identification number, and influence matrix files because the program issues its own DDEF commands to TSS.

USER: CALL INF\$

PROG: ENTER BATCH

USER: Enter carriage return for conversational mode.

PROG: ENTER ID1.

USER: Enter identification number of the geometry file. If a negative number is entered, then the most recent geometry file will be used. Enter zero to terminate execution.

PROG: +



## Influence Matrix Program

USER: At this point the user must begin entering commands. After each command he should enter the input, if any, associated with that command. The program does not prompt for this input. If the command is any of the commands except STOP, START, or WING, the program will respond with another + sign and the user must enter the next command. With a STOP command, the program will terminate. With a START command the program will request a value for ODISK. If output on the terminal is desired, enter zero. Otherwise enter an integer from 1-9. The program will respond by giving the name of the file where the output will be found. Then the program will commence with the calculation of the influence matrix. This may take many minutes on the AMES IBM 360/67 since the calculation can be lengthy. After the calculation has been finished, the program will respond with another + sign. At this time the user may make modifications to the data and calculate another influence matrix or else he may give the STOP command. After a WING command the program loops back to the point where it requests ID1. After a WING command has been given, the program resets variables which have default values. This means that, for example, if the user had given the NSTORE command once, he would have to give it again after a WING command to retain the effect of the NSTORE command.

### 3.3 TSS BATCH MODE

The batch mode operates the same as the conversational mode with the sole exception that a "T" must be put in column 1 on the first card (for the logical variable BATCH). This "T" suppresses all subsequent conversational prompts.

#### 4 DESCRIPTION OF COMMANDS

In all cases the first three letters of a command are sufficient input. All integer data should be entered in a 1615 format and all floating point data in 8F10.0 format. The input, if any, associated with each command is to be entered on the following line in conversational processing and in batch processing is to be on cards immediately following the command card. The same command may occur more than once. This is useful in correcting data entered in error.

Although the commands may, for the most part, be entered in any order the following command sequences must be observed: PP, if it is to be given, must precede the NMAX and CWTYPE commands. The START command must precede the WING and STOP commands. The MM command must precede the SWTYPE, KK, and MREF commands.

Certain commands reverse their effect each time the command is used. In other words, if the subject commands are given twice, the effect will be the same as if they were never given. Also the variables associated with these commands are reset after a WING command. This means that if, for example, the NOHEAD command were in effect, the user would have to give this command again following a WING command in order to retain the effect of the NOHEAD command. The commands which reverse their effect are: IWRITE, NSTORE, IWCCP, IWSEP, SUPPRESS, CCI, NOHEAD, and NINT.

Besides resetting the variables associated with the reversible commands, the WING command causes all other variables to be reset to their default values. These default values either come from the geometry file or are set by this influence matrix program.

The commands which are the most important and basic are: JJ, which adjusts the number of spanwise integration points; MM and SWTYPE, which adjust the spanwise control points; PP and SWTYPE, which adjust the chordwise control points; START, which initiates the major part of the computation; WING, which reads in geometry data for a new wing; and STOP, which terminates the execution. These and all of the other commands are explained fully below:

## Influence Matrix Program

### CCI

Effect: Causes chordwise convergence information to be printed. For a typical case this command may cause an additional 5000 lines of output, so this command is not generally used except when debugging.

### CONTINUE

Effect: Causes processing to continue in batch mode when an erroneous command is encountered. Otherwise execution will terminate. In the conversational mode an invalid command message will be issued and processing will continue.

### CWTYPE

Input: CWTYPE (integer variable)  
(CHICP(P),P=1,PP) (if CWTYPE<0).

Effect: Causes program to read CWTYPE, an integer denoting the type of chordwise control point distribution. (CHICP(P),P=1,PP) will then be computed if CWTYPE  $\geq 0$  or read if CWTYPE < 0. (CHICP) is an array of PP elements giving the relative chordwise location of the control points ( $0 \leq \text{CHICP} \leq 1$ ). The control points which are computed are the Wagner points if CWTYPE=0 (fig. 1) or the Multhopp points if CWTYPE>0 (fig. 2). The accuracy with which the downwash modes are calculated decreases for control points approaching but not on the leading and trailing edges. If this command is not given the default chordwise control points on the geometry file will be used. This command must be preceded by a PP command if it is necessary to give a PP command.

### DELTA0

Input: DELTA0

Effect: Causes program to read DELTA0. DELTA0 is a parameter in the spanwise integration described in ref. 1. It has a default value

## Influence Matrix Program

of  $AR^{**2}$  where AR is the aspect ratio.

### EPS

Input: EPS

Effect: Causes program to read EPS, an integration parameter discussed in ref. 1. The default value for EPS is .5.

### IWCCP

Effect: Causes printing of chordwise control points. This printing will not occur until after the START command has been given.

### IWRITE

Effect: Causes printing of the influence matrix.

### IWSCP

Effect: Causes printing of spanwise control points. This printing will not occur until after the START command has been given.

### JJ

Input: JJ

Effect: Causes a new JJ to be read. JJ is the number of spanwise integration points. The default value is JJMAX, which is a number read from the geometry file. If necessary, JJ will be adjusted downwards until  $MOD(JJMAX+1, JJ+1)=0$ . This adjustment is necessary because the integration points used must be identical to or a symmetric subset of the integration points which are on the geometry file.

### KK

Input: KK

Effect: Causes new KK to be read. KK is the maximum order of the spanwise pressure modes. The actual number of modes computed is equal to KK

## Influence Matrix Program

unless  $MODES \leq 0$  (see below). This command must follow the MM command if  $KK \neq MM$ , otherwise this command is not needed.

### LMAX

Input: LMAX

Effect: Causes program to read LMAX, which is the maximum number of chordwise integration points which should be used (excluding the point on the leading edge). This number will be adjusted downwards until  $LMAX+1=(LMIN+1)*2**I$  where I is an integer. It will also be adjusted downwards until it is as small or smaller than LMXMX in subroutine INITAL. Currently LMXMX = 511. This relationship occurs because of the nesting of the chordwise quadrature points (ref. 1).

### LMIN

Input: LMIN

Effect: Causes program to read LMIN, which is the minimum number of chordwise integration points which should be used. If this command is not given then LMIN will be computed according to the type of chordwise control points which will be used. If  $CWTYPE < 0$ , then the program assumes  $LMIN=1$ . If  $CWTYPE \geq 0$ , then LMIN will be computed so as to align the chordwise control points with the chordwise integration points. The latter improves the convergence rate of the chordwise integration when the distance between the spanwise control and integration points is small.

### MACH

Input: MACH (floating point)

Effect: Causes program to read a new Mach number. The default value is the value stored on the geometry file.

### MM

## Influence Matrix Program

Input: MM

Effect: Causes program to read MM, which is the number of spanwise control points. The default value for MM is the value of NF stored on the geometry file. This command also sets KK equal to MM and MREF equal to MM, so, if the KK and/or MREF commands are to be given, they should follow this command.

### MODES

Input: MODES

Effect: Causes program to read MODES, which denotes the type of spanwise pressure modes to be used. The default value is 1, which causes both the symmetric and antisymmetric downwash modes to be calculated. If MODES<0 only the antisymmetric downwash modes will be calculated. If MODES=0 only the symmetric downwash modes will be calculated.

### MREF

Input: MREF

Effect: Causes program to read MREF, which is a reference number for the calculation of the spanwise integration points. MREF is the same as NN in the geometry program (ref. 2) and is explained therein. Also see the explanation of the SWTYPE command. If MREF=MM this command is not needed. If MREF≠MM then this command should be given after the MM command.

### NINT

Effect: Causes number of chordwise integration points for each control point and spanwise integration point to be printed out.

### NMAX

Input: NMAX

## Influence Matrix Program

**Effect:** Causes program to read NMAX, which is the number of chordwise pressure modes. The default value for NMAX is PP. If NMAX $\neq$ PP, this command must follow the PP command. If NMAX=PP, this command is not necessary.

### NOHEAD

**Effect:** Suppresses the case information which is ordinarily printed after ODISK is entered and just prior to computing the influence matrix. This command is generally only used in the conversational mode when the output is at the terminal and then it serves to decrease terminal time.

### NSTORE

**Effect:** Causes the program not to create an influence matrix file. This command is generally only used when testing the program and frequently is used with the IWRITE command.

### PP

**Input:** PP

**Effect:** Causes program to read PP, which is the number of chordwise control stations. This command also sets NMAX equal to PP, so if the NMAX command is to be given, it should be given after the PP command.

### START

**Effect:** Causes the program to print a page of heading material containing the integration parameters and other information and then starts computation of Aerodynamic Influence Matrix. As the computation is being done the influence matrix is stored PP rows at a time (if the NSTORE command has not been given).

### STOP

**Effect:** Terminates execution.

## Influence Matrix Program

### SUPPRESS

Effect: Causes suppression of warning messages occurring because the chordwise integration fails to converge.

### SWTYPE

Input: SWTYPE (integer)  
(NINDEX(M), M=1, MM) (if SWTYPE $\neq$ 0)

Effect: Causes program to read SWTYPE, which is an integer denoting the type of spanwise control points. If SWTYPE=0, then the usual Multhopp distribution will be used. If SWTYPE $\neq$ 0, then the array (NINDEX) will be read and the spanwise control points will be given by  $\text{COS}(\text{NINDEX}(M) * 3.14159... / (\text{MREF} + 1))$  for  $M=1, 2, \dots, \text{MM}$ . This command should follow the MM command, if the MM command should be given.

### TSS

Input: A TSS command of 80 characters or less.

Effect: The command is passed to the AMES' TSS operating system. After the system processes the command, control returns to the program. This command is a special one for the AMES' TSS version of the program.

### WING

Effect: Causes program to request a new value for ID1 so that another wing can be processed. All variables and commands are reinitialized after this command is given.



## Influence Matrix Program

### 5 SAMPLE TERMINAL SESSION

A sample terminal session on the AMES' 360/67 TSS computer system is given in this section with additional comments added in parenthesis. During this session 3 aerodynamic influence matrices for an AR = 2 wing were computed.

For the first influence matrix the default control points and integration stations were changed. For the second influence matrix everything was the same as for the first except that the Mach number was changed. Both matrices were printed out, but not stored.

For the third matrix the default integration stations and control points were used. The defaults were easily restored by using the WING command. This matrix was computed and stored, but not printed because of its large size.

The following example was reproduced from an actual terminal session:

```
LOGON userid,password,terminal id
AMES USYSLIB
JOBLIBS SYSULIB
JBLB MEDAN
CALL INF$
(The influence matrix program is now in control.)
ENTER BATCH
(carriage return)
ENTER ID1
    4
+
IWRITE
+
JJ
    47
+
PP
    3
+
CWTYPE
    0
+
MM
    3
```

# Influence Matrix Program

```
+
IWCCP
+
IWSCP
+
NSTORE
+
TSS
CPUTIME?
  0.765 SECONDS
+
START
ENTER ODISK
  1
OUTPUT IS ON ...OUTPUT.AIM.N1...
ID2= 0
(Now the program prints output and computes the
influence matrix.)
+
TSS
CPUTIME?
  2.978 seconds
(By subtracting the previous CPUTIME from the one
above, the user may determine how many CPU seconds it
took to calculate and store the influence matrix
program.)
+
MACH
0.8
+
START
ENTER ODISK
  2
OUTPUT IS ON ...OUTPUT.AIM.N2...
ID2= 0
+
TSS
CPUTIME?
  4.944 seconds
+
WING
ENTER ID1
  4
+
IWCCP
+
IWSCP
```

# Influence Matrix Program

```
+
TSS
CPUTIME?
  5.355 seconds
+
START
ENTER ODISK
  3
OUTPUT IS ON ...OUTPUT.AIM.N3...
ID2= 5
(The influence matrix will be on the file named
AIM.X004.X005 .)
+
TSS
CPUTIME?
  59.450 SECONDS
+
STOP
TERMINATED: STOP
(The operating system is now in control.)
PRINT OUTPUT.AIM.N1,PRTSP=EDIT,STATION=RMT05
  PRINT BSN=????, ??? LINES
PRINT OUTPUT.AIM.N2,PRTSP=EDIT,STATION=RMT05
  PRINT BSN=????, ??? LINES
PRINT OUTPUT.AIM.N3,PRTSP=EDIT,STATION=RMT05
  PRINT BSN=????, ??? LINES
LOGOFF
```

## Influence Matrix Program

### 6 INPUT FILES

The following disk files are read by the program. The Ames' TSS version of the program issues its own DDEF commands for the files, so none need be given. For other systems appropriate control cards will have to be supplied for units 7 and 9.

#### 6.1 GEOMETRY FILE

This file is a variable record length file and is read from unit 7.

The first record contains identification and title information including the number of control points and integration points.

The next record contains the chordwise control points, the array of indices from which the spanwise control points are derived, the tangents of the wing edge sweep angles at the integration stations, etc. For a complete description of this file see Ref. 2.

On the AMES' TSS system this file has the name GEOM.XI where I is the numerical value of ID1.

#### 6.2 IDENTIFICATION NUMBER FILE

This file is read from unit 9 and rewritten on unit 9 and contains identification numbers in binary form. The second number on this file (ID2) is incremented by 1 and then the file is rewritten using the incremented value of ID2. ID2 is the identification number which will be used for the aerodynamic influence matrix file. It is printed out with the heading information for each case.

On the AMES' TSS system this file has the name IDFILE.

## Influence Matrix Program

### 7 INFLUENCE MATRIX (AIM) FILE

This file is a variable record length file and is written on unit 11. The Ames' TSS version of the program issues its own DDEF command for this file, so none need be given. For another system appropriate control cards would have to be supplied for unit 11.

The first record contains identification and title information plus information about the size of the matrix and location of spanwise and chordwise control points.

The second and subsequent records contain the influence matrix itself.

On the AMES' TSS system this file has the name AIM.XI.XJ where I is the numerical value of ID1 and J is the numerical value of ID2.

The specific information stored on the file is the following:

#### FIRST LOGICAL RECORD

ID1	Identification number of the geometry file.
ID2	Identification number of AIM file.
NSEQ	This integer no longer serves a practical purpose.
(TITLE)	Titling information of up to 80 characters. This array is 26 integer words long.
NTITL	Number of words in TITLE to be printed.
PP	Number of chordwise control stations (integer).
CWTYPE	An integer denoting the type of chordwise control point distribution.
MM	Number of spanwise control points.
MREF	A reference number for the calculation of the spanwise integration points.

## Influence Matrix Program

SWTYPE    An integer denoting the type of spanwise control points.

NMAX      Number of chordwise pressure modes.

KK        Maximum order of the spanwise pressure modes.

MODES     Denotes the type of spanwise pressure modes whose downwash modes have been calculated and are stored on subsequent records of this file.

UNSYM     Integer which, when not equal to 0, indicates an unsymmetric wing.

LMIN      Minimum number of chordwise integration points which was used.

LMAX      Maximum number of chordwise integration points which was used.

JJMAX     Maximum number of spanwise integration stations.

JJ        Number of integration points which have been used.

MACH      Mach number.

EPS       An integration parameter.

DELTA0    A parameter in the spanwise integration.

(CHICP)   An array containing PP chordwise stations at which the downwash modes have been determined.

(NINDEX)   An array containing MM integers for computing the spanwise control point positions.

(ETACP)   MM spanwise stations at which the downwash modes have been determined. Although (ETACP) can be derived from (NINDEX) and MREF, it was considered convenient to store these points despite the redundancy.

## Influence Matrix Program

### NEXT MMP RECORDS

((ALFA(N,K,P),N=1,NMAX),K=KL,KK,KJUMP),P=1,PP)

Where  $KL=1$  if  $MODES \geq 0$  or  $KL=2$  if  $MODES < 0$  and  $KJUMP=1$  if  $MODES > 0$  or  $KJUMP=2$  if  $MODES \leq 0$  and  $MMP=MM$  if the wing is unsymmetrical, or  $MMP=(MM+1)/2$  if the wing is symmetrical and  $MODES \geq 0$  or  $MMP=MM/2$  if the wing is symmetrical and  $MODES < 0$ .

## Influence Matrix Program

### 8 REFERENCES

1. Medan, R. T.: Improvements to the Kernel Function Method of Steady, Subsonic Lifting Surface Theory. NASA Rept. No. TMX-62,327, March 1974.
2. Medan, R. T.: Geometry Program for Aerodynamic Lifting Surface Theory. NASA Rept. No. TMX-62,309, September 1973.
3. Lemmer, O. J., and Medan, R. T.: Equation Solving Program for Aerodynamic Lifting Surface Theory. NASA Rept. No. TMX-62,325, January 1974.
4. Medan, R. T., and Ray, K. S.: Boundary Condition Program for Aerodynamic Lifting Surface Theory, NASA Rept. No. TMX-62,323, December 1973.
5. Medan, R. T., and Ray, K. S.: Plotting Program for Aerodynamic Lifting Surface Theory. NASA Rept. No. TMX-62,321, November 1973.



## FIGURES

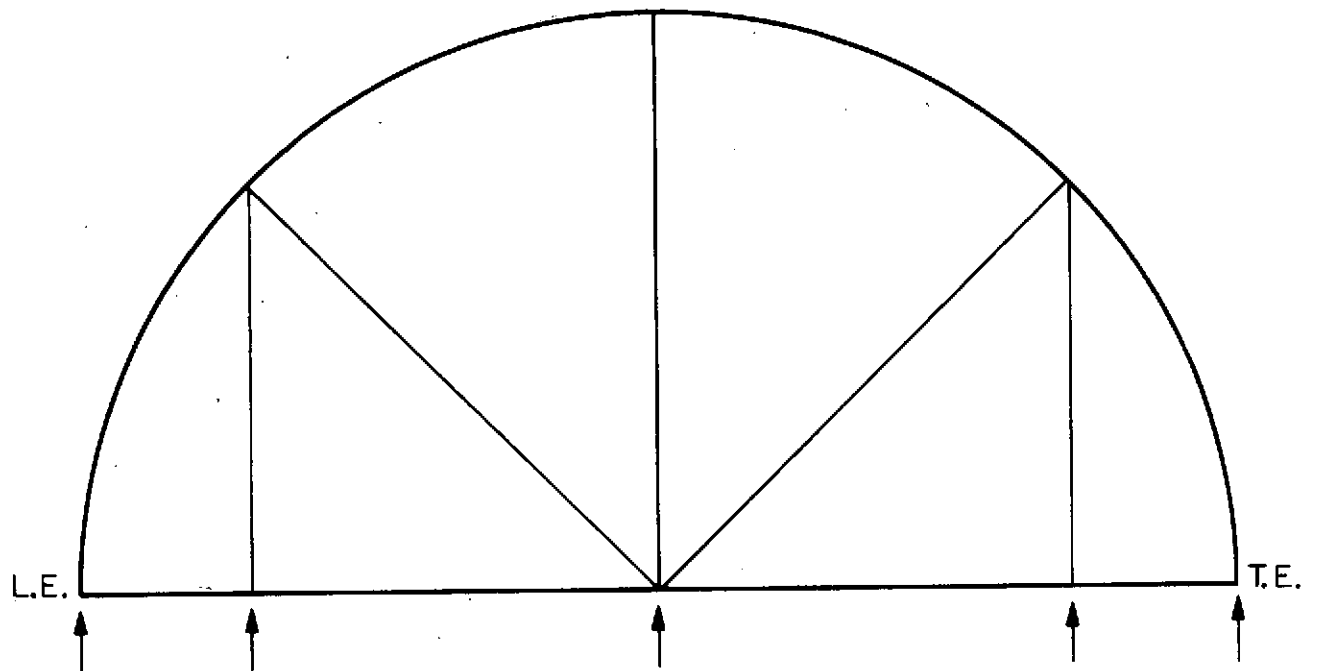


Figure 1.- The Wagner (CWTYPE = 0) chordwise control point distribution for PP

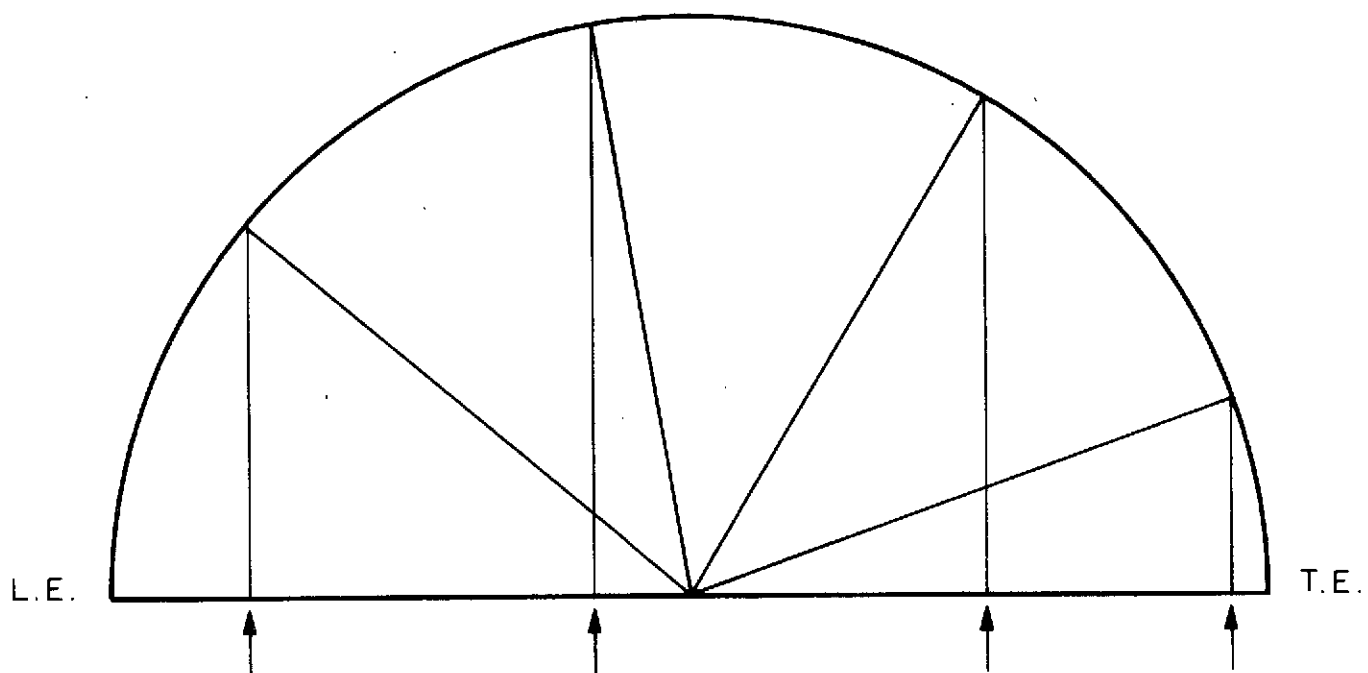


Figure 2.- The MULTHOPP (CWTYPE > 0) chordwise control point distribution for PP = 4.

## APPENDIX-I

### OUTPUT FROM SAMPLE SESSION

OUTPUT.AIM.N1  
OUTPUT.AIM.N2  
OUTPUT.AIM.N3

# DETERMINATION OF AERODYNAMIC INFLUENCE MATRIX

TITLE = RECTANGULAR WING AR = 2 11-13-73

ID1 = 4  
 ID2 = 0  
 NSEQ = 1  
 PP = 3  
 GWTYPE = 0  
 MM = 3  
 MREF = 3  
 SWTYPE = 0  
 NMAX = 3  
 KK = 3  
 MODES = 1  
 UNSYM = 0  
 LMIN = 1  
 LMAX = 511  
 JJMAX = 191  
 JJ = 47  
 DELTA0 = 4.00000  
 EPS = 0.50000  
 MACH = 0.00000  
 INFST0 = 1  
 (CHICP)

0.000000

0.500000

1.000000

1 INDEX ETA

1 1 0.70711

2 2 -0.00000

3 3 -0.70711

STARTING 2000 M LOOP

ETA= 0.7071068 CHI=0.0000000 #INT.PTS.= 260  
 (ALFA(N, 1, 1),N=1,NMAX)  
 0.7742263 -0.3409374 -0.4197662  
 (ALFA(N, 2, 1),N=1,NMAX)  
 1.5218760 -0.3566590 -0.5357901  
 (ALFA(N, 3, 1),N=1,NMAX)  
 1.3341286 -0.1946583 -0.3357640

ETA= 0.7071068 CHI=0.4999998 #INT.PTS.= 372  
 (ALFA(N, 1, 2),N=1,NMAX)  
 1.1218965 0.2500033 0.5498438  
 (ALFA(N, 2, 2),N=1,NMAX)  
 2.5702220 0.7071112 0.9359384  
 (ALFA(N, 3, 2),N=1,NMAX)  
 2.5568092 0.7500026 0.8005121

ETA= 0.7071068 CHI=1.0000000 #INT.PTS.= 260  
 (ALFA(N, 1, 3),N=1,NMAX)  
 1.2795500 0.8409446 -0.4197660  
 (ALFA(N, 2, 3),N=1,NMAX)  
 3.0584338 1.7708834 -0.5357904  
 (ALFA(N, 3, 3),N=1,NMAX)  
 3.1051411 1.6946648 -0.3357644

ETA=-0.0000000 CHI=0.0000000 #INT.PTS.= 256  
 (ALFA(N, 1, 1),N=1,NMAX)  
 1.0276747 -0.4824353 -0.6103635  
 (ALFA(N, 2, 1),N=1,NMAX)  
 -0.0000000 0.0000000 0.0000000  
 (ALFA(N, 3, 1),N=1,NMAX)  
 -1.5837338 0.3379765 0.5291822

FIA=-0.0000000 CHI=0.4999998 #INT.PTS.= 272  
 (ALFA(N, 1, 2),N=1,NMAX)  
     1.2456775      0.2500031      0.6963909  
 (ALFA(N, 2, 2),N=1,NMAX)  
     0.0000000      0.0000000      0.0000000  
 (ALFA(N, 3, 2),N=1,NMAX)  
     -2.6826989      -0.7500027      -0.9502366

ETA=-0.0000000 CHI=1.0000000 #INT.PTS.= 256  
 (ALFA(N, 1, 3),N=1,NMAX)  
     1.3807179      0.9824502      -0.6103632  
 (ALFA(N, 2, 3),N=1,NMAX)  
     0.0000000      0.0000000      0.0000000  
 (ALFA(N, 3, 3),N=1,NMAX)  
     -3.2069144      -1.8379766      0.5291820

INT. POINT TOTAL = 1676

# DETERMINATION OF AERODYNAMIC INFLUENCE MATRIX

TITLE = RECTANGULAR WING AR = 2 11-13-73

ID1 = 4  
 ID2 = 0  
 NSEQ = 2  
 PP = 3  
 CWTYPE = 0  
 MM = 3  
 WREF = 3  
 SVTYPE = 0  
 NMAX = 3  
 KK = 3  
 MODES = 1  
 UNSYM = 0  
 LMIN = 1  
 LMAX = 511  
 JJMAX = 191  
 JJ = 47  
 DELTA0 = 4.00000  
 EPS = 0.50000  
 MACH = 0.80000  
 INFSTU = 1

(CHICP)

0.000000

0.500000

1.000000

I	INDEX	ETA
---	-------	-----

1	1	0.70711
---	---	---------

2	2	-0.00000
---	---	----------

3	3	-0.70711
---	---	----------



STARTING 2000 M LOOP

ETA= 0.7071068 CHI=0.0000000 #INT.PTS.= 344  
(ALFA(N, 1, 1),N=1,NMAX)  
0.5544372 -0.1843266 -0.2398086  
(ALFA(N, 2, 1),N=1,NMAX)  
1.1436427 -0.1759124 -0.2858456  
(ALFA(N, 3, 1),N=1,NMAX)  
1.0174537 -0.0916097 -0.1686799

ETA= 0.7071068 CHI=0.4999998 #INT.PTS.= 408  
(ALFA(N, 1, 2),N=1,NMAX)  
0.9741887 0.2500034 0.3921047  
(ALFA(N, 2, 2),N=1,NMAX)  
2.4197948 0.7071113 0.7525932  
(ALFA(N, 3, 2),N=1,NMAX)  
2.4888277 0.7500026 0.7003559

ETA= 0.7071068 CHI=1.0000000 #INT.PTS.= 344  
(ALFA(N, 1, 3),N=1,NMAX)  
1.1422063 0.6843331 -0.2398084  
(ALFA(N, 2, 3),N=1,NMAX)  
2.9322974 1.5901354 -0.2858455  
(ALFA(N, 3, 3),N=1,NMAX)  
3.0470359 1.5916157 -0.1686796

ETA=-0.0000000 CHI=0.0000000 #INT.PTS.= 336  
(ALFA(N, 1, 1),N=1,NMAX)  
0.7352671 -0.2561758 -0.3469462  
(ALFA(N, 2, 1),N=1,NMAX)  
-0.0000000 0.0000000 0.0000000  
(ALFA(N, 3, 1),N=1,NMAX)  
-1.1952249 0.1642214 0.2773985

ETA=-0.0000000 CHI=0.4999999 #INT.PTS.= 400  
 (ALFA(N, 1, 2),N=1,NMAX)  
   1.0284566      0.2500033      0.4638452  
 (ALFA(N, 2, 2),N=1,NMAX)  
   0.0000000      0.0000000      0.0000000  
 (ALFA(N, 3, 2),N=1,NMAX)  
   -2.5433286      -0.7500028      -0.7731673

ETA=-0.0000000 CHI=1.0000000 #INT.PTS.= 320  
 (ALFA(N, 1, 3),N=1,NMAX)  
   1.1878049      0.7561866      -0.3469412  
 (ALFA(N, 2, 3),N=1,NMAX)  
   0.0000000      0.0000000      0.0000000  
 (ALFA(N, 3, 3),N=1,NMAX)  
   -3.0928598      -1.6642203      0.2773957

INT. POINT TOTAL = 2152

# DETERMINATION OF AERODYNAMIC INFLUENCE MATRIX

TITLE = RECTANGULAR WING AR = 2 11-13-73

ID1 = 4  
 ID2 = 2  
 NSEQ = 1  
 PP = 5  
 CWTYPE = 0  
 MM = 11  
 MREF = 11  
 SWTYPE = 0  
 NMAX = 5  
 KK = 11  
 MODES = 1  
 UNSYM = 0  
 LMIN = 1  
 LMAX = 511  
 JJMAX = 191  
 JJ = 191  
 DELTA0 = 4.00000  
 EPS = 0.50000  
 MACH = 0.00000  
 INFSTO = 0

(CHICP)

0.000000  
 0.146447  
 0.500000  
 0.853553  
 1.000000

I	INDEX	ETA
1	1	0.96593
2	2	0.86603
3	3	0.70711
4	4	0.50000
5	5	0.25882
6	6	-0.00000
7	7	-0.25882
8	8	-0.50000
9	9	-0.70711
10	10	-0.86603
11	11	-0.96593

STARTING 2000 M LOOP

ROWS 1- 5 HAVE BEEN STORED

ROWS 6-10 HAVE BEEN STORED

ROWS 11-15 HAVE BEEN STORED

ROWS 16-20 HAVE BEEN STORED

ROWS 21-25 HAVE BEEN STORED

ROWS 26-30 HAVE BEEN STORED

INT. POINT TOTAL = 59912

**APPENDIX-II**  
**PROGRAM LISTING**

C

```

DOUBLE PRECISION ETA, PI, STHETA, SINES, CSINES, ALFA
INTEGER PP, CHTYPE, SHTYPE, TITLE, UNSYM, U5, U6, U7, U20, U11, U12
INTEGER ODISK, UCO
LOGICAL CONTIN, SUPRES, NOCCI, NINT
LOGICAL FRESH, CONVR5, BATCH
REAL IRREG, IRREGI, MACH
REAL IK
REAL JJS, MODS, MMS, MRES, NHAS, LMAS, LMIS, KKS, NSTS, IHR5,
1IWCS, IWSS, MACS, NMS, NINS
DIMENSION TITLE(26), CHICP(20), NINDEX(47)
DIMENSION ETA(384), STHETA(384), XSILIP(383), CORDIP(383)
DIMENSION TLEL(383), TLER(383), TTEL(383), TTER(383)
C..... DIMENSION IRREG AT LEAST AS GREAT AS 6*NMAX*KK.
C..... DIMENSION IRREGI AT LEAST AS GREAT AS NMAX*PP.
C..... DIMENSION ALFA AT LEAST AS GREAT AS NMAX*KK*PP.
C..... DIMENSION ALFA2 AT LEAST AS GREAT AS NMAX*KK*PP
DIMENSION IRREG(1800), IRREGI(225)
DIMENSION ALFA(2000)
DIMENSION ALFA2(2000)
DIMENSION IK(47)
DIMENSION SK(47)
DIMENSION GNL(10)
DIMENSION GPNL(10)
DIMENSION GNT(10)
DIMENSION DMNDX(200)
DIMENSION HNYZRO(200)
DIMENSION SINES(48)
DIMENSION CSINES(48)
DIMENSION LTOTP(20)
DIMENSION NINTP(20)
DIMENSION G(5120), XP(5120), HHN(10)
DIMENSION ETACP(47)
COMMON PI
COMMON LMAX, LPWR, LMIN, LMAX1, G, XP
COMMON X, YS, DELTA, HHN, NMAX, SUPRES, NOCCI, RLLMIN, U6, LTOTAL
EQUIVALENCE (ALFA2, IRREG, ETACP)
DATA U5, U7, U20, U11/5, 7, 9, 11/ . UCO /6/
DATA COM /4H /

```

34

DATA WINS / 3HWIN /  
 DATA CONS / 1HCON /  
 DATA STOS / 3HSTO /  
 DATA DELS / 3HDEL /  
 DATA JJ S / 3HJJ /  
 DATA MODS / 3HMOD /  
 DATA PP S / 3HPP /  
 DATA CWTs / 3HCWT /  
 DATA MM S / 3HMM /  
 DATA MRES / 3HMRE /  
 DATA SWTs / 3HSWT /  
 DATA NMAS / 3HNSA /  
 DATA LMAS / 3HLMA /  
 DATA LMIS / 3HLMI /  
 DATA KK S / 3HKK /  
 DATA NSTs / 3HNST /  
 DATA IWRs / 3HIWR /  
 DATA IWCS / 3HIWC /  
 DATA INSS / 3HIWS /  
 DATA MACs / 3HMAC /  
 DATA SUPs / 3HSUP /  
 DATA CCIS / 3HCCI /  
 DATA NOHS / 3HNOH /  
 DATA NINS / 3HNIN /  
 DATA STAS / 3HSTA /  
 DATA TSSs / 3HTSS /  
 PI=3.141592653589793 DO

U6 = UCO

C.....CONVRS=.TRUE. FOR CONVERSATIONAL USE. SET CONVRS TO

C.....FALSE, FOR BATCH.

WRITE(U6,2024)

READ(U5,6) BATCH

CONVRS=.NOT.BATCH

20 CONTINUE

IF(CONVRS) WRITE(UCO,2028)

READ(U5,5) ID1

C

C.....AIMFIL IS FOR AMES; TSS VERSION ONLY. AIMFIL ISSUES DDEF COMMANDS TO TSS.

C

```

CALL GEMFIL(ID1)
REWIND U7
READ(U7)ID1,PP,MM,CWTYPE,SWTYPE,UNSYM,NDL,NDT,MREF,JJMAX,I,TITLE,N=
1TITL
  READ(U7)(CHICP(I),I=1,PP),(NINDEX(I),I=1,MM),
1(TLER(J),J=1,JJMAX),(TLER(J),J=1,JJMAX),
1(TTEL(J),J=1,JJMAX),(TTER(J),J=1,JJMAX),
1(ETA(J),J=1,JJMAX),(STHETA(J),J=1,JJMAX),
1(XSILIP(J),J=1,JJMAX),(CORDIP(J),J=1,JJMAX),
1DUM,DUM,AR,DUM,RMACH
  ENDFILE U7
35 CONTINUE
  NMAX=PP
  KK=MM
  FPS=.5
  DELTA0=AR**2
  JJ=JJMAX
  MODES=1
  NSEQ=0
  LMAX=0
  LMIN=0
  INFSTO=0
  IWRITE=0
  IWCCP=0
  IWSCP=0
  CONTIN=CONVRS
  SUPRES=.FALSE.
  NOCCJ=.TRUE.
  NOHEAD=0
  NINT=.FALSE.
  JJMAX1=JJMAX+1
  JJ1=JJMAX1
  MREF1=MREF+1
  MACH=RMACH
  MCALC=1
50 CONTINUE
  IF(CONVRS) WRITE(UCO,3)
  READ(US,1) COM
  IF (COM,EQ,WINS) GO TO 20

```



```

IF (COM, EQ, CONS) GO TO 80
IF (COM, EQ, STOS) GO TO 100
IF (COM, EQ, DELS) GO TO 110
IF (COM, EQ, EPSs) GO TO 115
IF (COM, EQ, JJ s) GO TO 120
IF (COM, EQ, MODs) GO TO 130
IF (COM, EQ, PP s) GO TO 140
IF (COM, EQ, CWTs) GO TO 150
IF (COM, EQ, MM s) GO TO 160
IF (COM, EQ, MRES) GO TO 170
IF (COM, EQ, SWTs) GO TO 180
IF (COM, EQ, NMAs) GO TO 190
IF (COM, EQ, LMAs) GO TO 200
IF (COM, EQ, LMIs) GO TO 210
IF (COM, EQ, KK s) GO TO 220
IF (COM, EQ, NSTs) GO TO 230
IF (COM, EQ, IWRs) GO TO 240
IF (COM, EQ, IWSs) GO TO 250
IF (COM, EQ, IWSs) GO TO 260
IF (COM, EQ, MACs) GO TO 270
IF (COM, EQ, SUPs) GO TO 280
IF (COM, EQ, CCIs) GO TO 290
IF (COM, EQ, NOHs) GO TO 295
IF (COM, EQ, NINs) GO TO 296
IF (COM, EQ, STAs) GO TO 300

```

37

C  
C.....FOR AMES: TSSS VERSION ONLY, ALWS TSS COMMAND TO BE ENTERED  
C

```

IF (COM, EQ, TSSs) GO TO 60
IF (CONVRS) WRITE (U5, 2023) COM
IF (CONVRS) GO TO 50
WRITE (U6, 2023) COM
STOP 13
60 CONTINUE
READ (U5, 7) (ETACP(I), I=1, 20)
CALL OBEY(80, ETACP)
GO TO 50
80 CONTINUE, NOT, CONTIN
GO TO 50

```

100 CONTINUE  
 STOP  
 110 READ (U5,4) DELTA0  
 GO TO 50  
 115 READ(U5,4) EPS  
 GO TO 50  
 120 READ(U5,5) JJ  
 JJ=MIN0(JJ,JJMAX)  
 JJ1=JJ+1  
 DO 122 I=1,JJMAX  
 IF(MOD(JJMAX1,JJ1),EQ.0) GO TO 123  
 122 JJ1=JJ1+1  
 123 JJ=JJ1-1  
 GO TO 50  
 130 READ(U5,5) MODES  
 GO TO 50  
 140 READ(U5,5) PP  
 NMAX=PP  
 GO TO 50  
 150 READ(U5,5) CWTYP  
 IF(CWTYP)151,152,157  
 151 READ(U5,4)(CHICP(I),I=1,PP)  
 GO TO 50  
 152 IF(PP=2)156,155,153  
 153 IMAX=PP-1  
 DTHETA=PI/FLOAT(PP-1)  
 DO 154 I=2,IMAX  
 154 CHICP(I)=(1.-COS(FLOAT(I-1)\*DTHETA))/2.  
 155 CHICP(PP)=1.  
 156 CHICP(1)=0  
 GO TO 50  
 157 DTHETA=PI/FLOAT(PP+PP+1)  
 DO 158 I=1,PP  
 158 CHICP(I)=(1.-COS(FLOAT(I+I)\*DTHETA))/2.  
 GO TO 50  
 160 READ(U5,5) MM  
 KK=MM  
 MREF=MM  
 MREF1=MREF+1

GO TO 50  
 170 READ(US,5) MREF  
     MREF=MAX0(1,MREF)  
     MREF=MIN0(MREF,JJ)  
     MREF1=MREF+1  
     DO 173 I=1,JJ  
         IF(MOD(JJ1,MREF1),EQ,0) GO TO 175  
 173 MREF1=MREF1-1  
 175 MREF=MREF1-1  
     GO TO 50  
 180 READ(US,5)SWTYPE  
     IF(SWTYPE) 182,185,182  
 182 READ(US,5)(NINDEX(1),I=1,MM)  
     GO TO 50  
 185 DO 187 I=1,MM  
 187 NINDEX(I)=I  
     GO TO 50  
 190 READ(US,5)NMAX  
     GO TO 50  
 200 READ(US,5)LMAX  
     GO TO 50  
 210 READ(US,5)LMIN  
     GO TO 50  
 220 READ(US,5)KK  
     GO TO 50  
 230 INFSTO=MOD(INFSTO+1,2)  
     GO TO 50  
 240 IWRITE=MOD(IWRITE+1,2)  
     GO TO 50  
 250 IWCCP=MOD(IWCCP+1,2)  
     GO TO 50  
 260 IWSCP=MOD(IWSCP+1,2)  
     GO TO 50  
 270 READ(US,4)MACH  
     GO TO 50  
 280 SUPRES=,NOT,SUPRES  
     GO TO 50  
 290 NOCCI=,NOT,NOCCI  
     GO TO 50

```

295 NOHEAD=MOD(NOHEAD+1,2)
    GO TO 50
296 NINT=,NOT,NINT
    GO TO 50
300 CONTINUE
    NSEQ=NSEQ+1
    ID2=0
    IF(INFSTO,NE,0)GO TO 303

```

```

C
C.....FOR AMES: TSS SYSTEM ONLY.  DEFINES DSNAME OF IDFILE.
    CALL OBEY(22,24HDDDEF FT09F001,,IDFILE

```

```

C
    REWIND U20
    READ(U20) ID, ID2,ID3,ID4
    ID2=ID2+1
    REWIND U20
    WRITE (U20) ID,ID2,ID3,ID4
    ENDFILE U20
    CALL OBEY(16,16HRELEASE FT09F001 )
    IF(NOHEAD,NE,0) WRITE(UCO,2027) ID2
40 303 CONTINUE
    IF(CONVRS) WRITE(UCO,2029)
    READ(US,5) ODISK
    IF(ODISK,EQ,0) GO TO 304
    U6=4
    ODISK=MOD(MAX0(1,ODISK),10)
    WRITE(UCO,2026) ODISK
    CALL OBEY (16,16HRELEASE FT04F001 )
    CALL CVRT(ODISK,1,
1 44H(1DDEF FT04F001,,OUTPUT,AIM,N,11.6X)
2ETACP,8,8H(8A4) )
    CALL OBEY (32,ETACP)
    REWIND U6
    IF(CONVRS)WRITE(UCO,2027) ID2
304 CONTINUE
    CALL INITAL(CWTYPE,PP)
    IF(NOHEAD,NE,0) GO TO 310
    WRITE(U6,2000)
    WRITE(U6,2001) (TITLE(N),N=1,NTITL)

```

```

WRITE(U6,2002) ID1,ID2,NSEQ
WRITE(U6,2003) PP,CWTYPE
WRITE(U6,2004) MM,MREF,SWTYPE
WRITE(U6,2005) NMAX,KK,MODES,UNSYM
WRITE(U6,2006) LMIN,LMAX
WRITE(U6,2007) JJMAX,JJ
WRITE(U6,2008) DELTA0,EPS,MACH
WRITE(U6,2009) INFSTO
IF(IWCCP.NE.0) WRITE(U6,2020) (CHICP(I),I=1,PP)
IF(IWSCP.EQ.0) GO TO 310
JRATIO=JJMAX1/MREF1
WRITE(U6,2021)
DO 305 I=1,MM
J=NINDEX(I)*JRATIO
ETAS=ETA(J)
305 WRITE(U6,2022) I,NINDEX(I),ETAS
310 CONTINUE
IF(MOD(JJMAX1,JJ1).NE.0) PAUSE'2'
IF(MOD(JJ1,MREF1).NE.0) PAUSE'3'
IF(INFSTO.NE.0) GO TO 340

```

C  
C.....COMPUTING SPANWISE CONTROL POINTS FOR AIM FILE: THE  
C.....ONLY PLACE ETACP IS USED IS IN WRITING THE INTRODUCTORY  
C.....RECORD OF THE AIM FILE.

```

JRATIO=JJMAX1/MREF1
DO 312 I=1,MM
J=NINDEX(I)*JRATIO
ETACP(I)=ETA(J)
312 CONTINUE

```

C  
C.....FOR AMES: TSS VERSION ONLY. AIMFIL ISSUES DDEF COMMANDS TO  
C.....THE TSS OPERATING SYSTEM.  
CALL AIMFIL(ID1,ID2)

C  
WRITE(U11) ID1,ID2,NSEQ,TITLE,NTITL,PP,CWTYPE,MM,MREF,SWTYPE,NMAX,  
1KK,MODES,UNSYM,LMIN,LMAX,JJMAX,JJ,MACH,EPS,DELTA0,  
1(CHICP(I),I=1,PP),(NINDEX(I),I=1,MM),  
1(ETACP(I),I=1,MM)  
340 CONTINUE

```

NMAXR=NMAX
MM1=MM+1
CALL INFMAT(MM,MREF,PP,NMAXR,UNSYM,KK,MODES,JJ,MCALC,JJMAX,INFSTO,=
1 IWRITE,US,U11,MACH,DELTA0,ETA,STHETA,XSILIP,CORDIP,CHICP,NINDEX,N1=
INT,IRREGI,IRREGI,ALFA,ALFA2,IK,SK,GNL,GPNL,GNT,DHNDX,HNYZRO,
1MM1,JJMAX1,SINES,CSINES,LTOTP,NINTP,EPS,TLEL,TLER,TTLE,ITER)
ENDFILE U11
GO TO 50
1 FORMAT(A3)
3 FORMAT(2H + )
4 FORMAT(8F10.0)
5 FORMAT(16I5 )
6 FORMAT(L1)
7 FORMAT(20A4)
2000 FORMAT(1H1,47H0DETERMINATION OF AERODYNAMIC INFLUENCE MATRIX. //)
2001 FORMAT(6H0TITLE,4X,4H= ,20A4)
2002 FORMAT(4H0ID1,6X,1H=,15/4H ID2,6X,1H=,15/5H NSEQ,5X,1H=,15)
2003 FORMAT(3H PP,7X,1H=,15/7H CWTYPE,3X,1H=,15)
2004 FORMAT(3H MM,7X,1H=,15/5H MREF,5X,1H=,15/7H SWTYPE,3X,1H=,15)
42 2005 FORMAT(5H NMAX,5X,1H=,15/3H KK,7X,1H=,15/6H MODES,4X,1H=,15/6H UNS.
1YM,4X,1H=,15)
2006 FORMAT(5H LMIN,5X,1H=,15/5H LMAX,5X,1H=,15)
2007 FORMAT(6H JJMAX,4X,1H=,15/3H JJ,7X,1H=,15)
2008 FORMAT(7H DELTA0,3X,1H=,F11.5/4H EPS,6X,1H=,F11.5/
15H MACH,5X,1H=,F11.5)
2009 FORMAT(7H INFSTO,3X,1H=,15)
2020 FORMAT(8H (CHICP)/(1X,F10.6))
2021 FORMAT(19H I INDEX ETA )
2022 FORMAT(1X,14,I5,F10.5)
2023 FORMAT(24H UNRECOGNIZED COMMAND: ,A3)
2024 FORMAT(13H ENTER BATCH )
2025 FORMAT(13H ENTER ODISK )
2026 FORMAT(29H OUTPUT IS ON ...OUTPUT,AIM,N ,11,3H,...)
2027 FORMAT(5H ID2=, I3)
2028 FORMAT(13H ENTER ID1 )
END
SUBROUTINE INFMAT(MM,MREF,PP,NMAXR,UNSYM,KK,MODES,JJ,MCALC,JJMAX,I
INFSTO,IWRITE,US,U11,MACH,DELTA0,ETA,STHETA,XSILIP,CORDIP,CHICP,NIN
IDEX,NINT,IRREGI,IRREGI,ALFA,ALFA2,IK, SK, GNL, GPNL, GNT,

```

1DHNDX, HNYZRO, MM1, JJMAX1, SINES, CSINES, LTOTP, NINTP, EPS,  
 1TLEL, TLER, TTEL, TTER)  
 DOUBLE PRECISION ETA, PI, STHETA, SINES, CSINES  
 DOUBLE PRECISION DK, DK2, DELTA, DELTA2, BJM, RJJ1, Y, ALFA, RJJ14  
 INTEGER PP, CWTYP, SWTYP, UNSYM, U5, U6, P, PL, PU, TJRATO, TJJMX1, TJX1M1,  
 1U11, GTOTAL  
 LOGICAL SYM, ASYM, TLE, TTE, TINT, NOTL, NOTT, NOTI, MNZERO, NE1, SUPRES, NOC  
 1CI, NINT  
 REAL IK, IRREG, IRREGI, MACH, JOLM, JOLP, JILM, JILP, J1TM, J1TP  
 DIMENSION NINDEX(MM)  
 DIMENSION CHIC(PP)  
 DIMENSION ETA(JJMAX1)  
 DIMENSION STHETA(JJMAX1)  
 DIMENSION XSILIP(JJMAX)  
 DIMENSION CORDIP(JJMAX)  
 DIMENSION TLEL(JJMAX), TLER(JJMAX), TTEL(JJMAX)  
 DIMENSION TTER(JJMAX)  
 DIMENSION IRREG(2,3,NMAXR, KK)  
 DIMENSION IRREGI(NMAXR, PP)  
 DIMENSION IK(MM)  
 DIMENSION SK(MM)  
 DIMENSION ALFA(NMAXR, KK, PP)  
 DIMENSION ALFA2(NMAXR, KK, PP)  
 DIMENSION GNL(NMAXR)  
 DIMENSION GPNL(NMAXR)  
 DIMENSION GNT(NMAXR)  
 DIMENSION DHNDX(NMAXR, PP)  
 DIMENSION HNYZRO(NMAXR, PP)  
 DIMENSION SINES(MM1)  
 DIMENSION CSINES(MM1)  
 DIMENSION LTOTP(PP)  
 DIMENSION NINTP(PP)  
 DIMENSION G(5120), XP(512), HHN(10)  
 COMMON PI  
 COMMON LMAX, LPWR, LMIN, LMAX1, G, XP  
 COMMON X, YS, DELTA, HHN, NMAX, SUPRES, NOCCI, RLLMIN, U6, LTOTAL  
 DATA C1/.6366197723/  
 DATA C2/-.3183098861/  
 DATA C3/1.273239545/

```

DATA C4/-.6366197723/
DATA C5/1.3333333/
DATA C6/.4501581579/
DATA C7/-.3001054386/
DATA C8/.125/
DATA C9/-.1.386294361/
DATA C10/-.8333333 E-1/
DATA C11/-.0625/
DATA C12/-.25/
JJ1=JJ+1
MRATIO=JJMAX1/(MREF+1)
JRATIO=JJMAX1/JJ1
TJRATO=JRATIO+JRATIO
ETA(JJMAX1)=-1.00
STHETA(JJMAX1)=0.00
C.....GTOTAL WILL BE THE GRAND TOTAL OF THE NUMBER OF
C.....INTEGRATION POINTS USED FOR ALL THE CONTROL POINTS,
GTOTAL = 0
IF (UNSYM) 110,100,110
100 SYM=.TRUE.
MMP=(MM+1)/2
IF(MODES.LT.0) MMP=MM/2
GO TO 120
110 SYM=.FALSE.
MMP=MM
120 ASYM=.NOT.SYM
KU2=MAX0(3,KK+1)
TJJMX1=JJMAX1+JJMAX1
TJX1M1=TJJMX1-1
C.....MODES.LT.0...ODD MODES
C.....MODES.EQ.0...EVEN MODES
C.....MODES.GT.0...ALL MODES
IF(MODES)130,140,150
130 KL=2
KJUMP=2
GO TO 160
140 KL=1
KJUMP=2
GO TO 160

```



```

150 KL=1
    KJUMP=1
160 CONTINUE
    MNZERO=MACH,NE,0.
    BETA=1.
    IF (MNZERO) BETA=SQRT(1.-MACH**2)
    TBETA=2.*BETA
    BETA2=BETA**2
    NE1=NMAX,EO,1
    TLE=CHICP(1),LT,1,E=6
    TTE=CHICP(PP),GT,.999999
    TINT=(PP,GT,2),OR. (PP,EQ,2 ,AND.((,NOT,TLE,OR.,NOT,TTE)),OR. ,NOT.
1 (TLE,OR,TTE)
    NOTL=,NOT,TLE
    NOTI=,NOT,TINT
    NOTT=,NOT,TTE
    PL=1.
    PU=PP
    IF (TLE) PL=2
    IF (TTE) PU=PP-1
45 C.....COMPUTING QUANTITIES ASSOCIATED WITH CONTROL POINTS
C.....ON LEADING AND TRAILING EDGES (EQS. 52,53,66,AND 67)
    SGN=1.
    GNL(1)=C1
    GPNL(1)=C2
    GNT(1)=C1
    IF (NE1) GO TO 250
    DO 200 N=2,NMAX
    GNL(N)=0
    GPNL(N)=C3*FLOAT(N-1)
    GNT(N)=SIGN(GPNL(N),SGN)
    SGN=-SGN
200 CONTINUE
250 CONTINUE
    IF (NOTI) GO TO 305
C.....COMPUTING QUANTITIES ASSOCIATED WITH INTERIOR CONTROL
C.....POINTS (EQS. 63 AND 64)
    DO 300 P=PL,PU
    X=2,*CHICP(P)=1.

```

```

PHI=ARCCOS(-X)
SPHI=SQRT(1.-X**2)
DHNDX(1,P)=C4*(1.-X)/SPHI**3
IF(NE1)GO TO 300
DO 290 N=2,NMAX
  DUM1=N-1
  DHNDX(N,P)=C1*DUM1*COS(DUM1*PHI)/SPHI
290 CONTINUE
300 CONTINUE
305 CONTINUE
RJJ1=JJ1
RJJ14=.25D0*RJJ1
DELCON=DELTA0/RJJ1**2
C.....DETERMINING HN OF EQ. 1 FOR THE CASE Y=0.
DO 350 P=1,PP
  X=2.*CHICP(P)-1.
  CALL INFZY(X,NMAX,MHN)
DO 350 N=1,NMAX
94 350 HNYZRO(N,P)=MHN(N)
  WRITE(U6,1)
C*****
DO 2000 M=MCALC,MMP
  M2=NINDEX(M)*MRATIO
C.....LTOTP(P)=NUMBER OF INTEGRATION POINTS USED OVER THE
C.....ENTIRE SPAN FOR EACH CHORDWISE CONTROL POINT LOCATION.
DO 360 P=1,PP
  LTOTP(P)=0
360 CONTINUE
  IF (NINT) WRITE(U6,7)M
  BB2C=BETA/CORDIP(M2)
  BBC=BB2C*BB2C
  ETAMS=ETA(M2)
  IF (NOTL) GO TO 500
C.....COMPUTING QUANTITIES ASSOCIATED WITH LEADING EDGE
C.....CONTROL POINTS AND WHICH VARY AS THE SPANWISE POSITION VARIES.
  TANLEL=TLEL(M2)
  COSPHI=BETA/SQRT(BETA2+TANLEL**2)
  EM=.5*(1.+TANLEL*COSPHI)
  EMCOMP=1.-EM

```

```

DUM1=SQRT(BB2C/COSPHI)
DUM13=DUM1**3
C.....EQS. 74 AND 75
JOLM=4.*DUM1*CEL2(EM,EM,0.)
JILM=C5*DUM13*CEL2(EM,EM,2.*EM*EMCOMP)
TANLER=TLER(M2)
IF(ABS(TANLEL+TANLER),LT,1.E-6) GO TO 400
COSPHI=BETA/SQRT(BETA2+TANLER**2)
EM=.5*(1.+TANLER*COSPHI)
EMCOMP=1.-EM
DUM1=SQRT(BB2C/COSPHI)
DUM13=DUM1**3
GO TO 410
400 CONTINUE
DUM2=EM
EM=EMCOMP
EMCOMP=DUM2
410 CONTINUE
JOLP=4.*DUM1*CEL2(EM,EM,0.)
JILP=C5*DUM13*CEL2(EM,EM,2.*EM*EMCOMP)
500 CONTINUE
IF(NOTL.AND,NOTT) GO TO 700
C.....EQ. 60
EM1=(1.+ETAMS)*.5
C.....EMICOM IS THE COMPLEMENTARY PARAMETER TO EM1.
EMICOM=(1.-ETAMS)*.5
DUM1=2.*EM1*EMICOM
C.....EQS. 58-59, ALSO SEE EQS. 76-77.
GAN3M=C6*CEL2(EM1,1.,1.-2.*FM1)
GAN1M=C7*CEL2(EM1,EM1,DUM1)
GAN3P=C6*CEL2(EMICOM,1.,1.-2.*EMICOM)
GAN1P=C7*CEL2(EMICOM,EMICOM,DUM1)
C.....COMPUTING QUANTITIES ASSOCIATED WITH TRAILING EDGE CONTROL
C.....POINTS AND WHICH VARY AS THE SPANWISE POSITION VARIES.
TANTEL=TYEL(M2)
COSPHI=BETA/SQRT(BETA2+TANTEL**2)
EM=.5*(1.+TANTEL*COSPHI)
EMCOMP=1.-EM
DUM1=SQRT(BB2C/COSPHI)**3

```

```

C.....EQ. 68
      J1TH=C5*DUM1*CEL2(EM,EM,2.*EM*EMCOMP)
      TANTER=TTER(M2)
      IF (ABS(TANTEL+TANTER).LT. 1.E-6) GO TO 600
      COSPHI=BETA/SQRT(BETA2+TANTER**2)
      EM=.5*(1.-TANTER*COSPHI)
      EMCOMP=1.-EM
      DUM1=SQRT(BB2C/COSPHI)**3
      GO TO 610
600 CONTINUE
      DUM2=EM
      EM=EMCOMP
      EMCOMP=DUM2
610 CONTINUE
      J1TP=C5*DUM1*CEL2(EM,EM,2.*EM*EMCOMP)
700 CONTINUE
C.....COMPUTING QUANTITIES ASSOCIATED WITH INTERIOR CONTROL POINTS.
C.....SINES AND COSINES IN EQ. 22-25
      J=(KL-1)*M2+TJXIM1
      DO 800 K=KL,KU2
      INDEX=MOD(J,JJMAX1)+1
      IF(MOD(J,TJJMX1)+1=INDEX)710,710,720
710 SINES(K)=STHETA(INDEX)
      CSINES(K)=ETA(INDEX)
      GO TO 730
720 SINES(K)=-STHETA(INDEX)
      CSINES(K)=-ETA(INDEX)
730 J=J+M2
C.....EQ. 10
800 SK(K)=STHETA(M2)*CSINES(K)
C.....COMPUTING IK OF EOS. 22-25
      DO 900 K=KL,KK,KJUMP
      IF(K=2) 810,820,830
810 IK(1)=C8*(CSINES(3)+C9)
      GO TO 900
820 IK(2)=C10*(2.D0*SINES(3)*SINES(2)+CSINES(3)*CSINES(2))
      GO TO 900
830 IF(K=3)840,840,850
840 IK(3)=C11*(1.5D0*SINES(4)*SINES(2)+.5D0*CSINES(4)*CSINES(2)+CSINES

```

```

1(3)+C9)
GO TO 900
850 DK=K
DK2=K*2
IK(K)=C12*((DK*SINES(K+1)*SINES(2)+CSINES(K+1)*CSINES(2))/(DK*DK-1
1,D0) = (DK2*SINES(K-1)*SINES(2)+CSINES(K-1)*CSINES(2))/(DK2*DK2-
11,D0))
900 CONTINUE
C.....EG. 87
BJM=RJJ14/STHETA(M2)
C.....COMPUTING THE EXACT INTEGRALS OF THE IRREGULAR
C.....PARTS AND THE CONTRIBUTION TO ALFA FROM ETAI=ETA.
IF (NOT1) GO TO 1000
DUM7=0.
DUM8=0.
IF (TANLEL,NE,TANTEL) DUM7=-.5*(TANLEL-TANTEL)/CORDIP(M2)
IF (TANLER,NE,TANTER) DUM8=-.5*(TANLER-TANTER)/CORDIP(M2)
DO 950 K=KL, KK, KJUMP
DUM1=J0LM*CSINES(K)
DUM2=J0LP*CSINES(K)
DUM6=FLOAT(K-1)*SINES(K)/STHETA(M2)
DUM3=J0LM*DUM6
DUM4=J1LM*CSINES(K)
DUM5=J0LP*DUM6
DUM6=J1LP*CSINES(K)
DO 950 N=1, NMAX
IRREGE(1,1,N,K)=GNL(N)*DUM1
IRREGE(2,1,N,K)=GNL(N)*DUM2
IRREGE(1,2,N,K)=GNL(N)*DUM3+GPNL(N)*DUM4+DUM7*IRREGE(1,1,N,K)
IRREGE(2,2,N,K)=GNL(N)*DUM5+GPNL(N)*DUM6+DUM8*IRREGE(2,1,N,K)
C.....EG. 51
ALFA(N,K,1)=IRREGE(1,1,N,K)*GAM3M+IRREGE(2,1,N,K)*GAM3P+IRREGE(1,2
1,N,K)*GAM1M+IRREGE(2,2,N,K)*GAM1P+BJM*SK(K)*HNYZRO(N,1)
950 CONTINUE
1000 IF (NOT1) GO TO 1100
BBC2 = BBC**2
DO 1050 P=PL, PU
DO 1050 N=1, NMAX
IRREGI(N,P)=.5*BBC2*DHNDX(N,P)

```

```

DO 1050 K=KL, KK, KJUMP
C.....EQ. 26
ALFA(N, K, P)=IRREGI(N, P)*IK(K)+BJM*SK(K)+HNYZRO(N, P)
1050 CONTINUE
1100 IF(NOTT) GO TO 1200
DO 1150 K=KL, KK, KJUMP
DUM1=-CSINES(K)*J1TH
DUM2=-CSINES(K)*J1TP
DO 1150 N=1, NMAX
IRREGI(1, 3, N, K)=DUM1*GNT(N)
IRREGI(2, 3, N, K)=DUM2*GNT(N)
C.....EQ. 65
ALFA(N, K, PP)=IRREGI(1, 3, N, K)*GAM1H+IRREGI(2, 3, N, K)*GAM1P+BJM*SK(K)
1*HNYZRO(N, PP)
1150 CONTINUE
1200 CONTINUE
JL=MOD(M2, TJRATO1)+JRATIO
C***** START OF SPANWISE INTEGRATION *****
DO 1600 J=JL, JJMAX, TJRATO
50 C.....DETERMINING SK(ETA), EQ. 19
L=(KL-1)*J+TJX1H
LJUMP=J*KJUMP
DO 1360 K=KL, KK, KJUMP
INDEX=MOD(L, JJMAX1)+1
SK(K)=ETA(INDEX)*STHETA(J)
IF(MOD(L, TJJMX1)+1.GT.INDEX) SK(K)=-SK(K)
1360 L=L+LJUMP
C.....EQ. 88
DETA=ETA(M2)-ETA(J)
DETA2=DETA*DETA
BJM=-STHETA(J)/(RJJ1+DETA2)
Y=TBETA/CORDIP(J)*DETA
YS=Y*Y
C.....LIMITING THE MINIMUM NUMBER OF CHORDWISE INTEGRATION
C.....STATIONS
RLLMIN=.46844/(YS+.00755)
C.....EQ. 90
ABDETA=DABS(DETA)
DELTA=DELCON/(SNGL(-BJM))*EPS

```

```

DUM2=2./CORDIP(J)
DUM1=DUM2*(XSILIP(M2)-XSILIP(J))=1.
DUM2=DUM2*CORDIP(M2)
SDETA=SQRT(ABDETA)*STHETA(J)
I=1
IF(DETA,LT,0,00)I=2
IF(NOTL) GO TO 1420
X=DUM1
CALL INFFUN
LTOTP(1)=LTOTAL+LTOTP(1)
IF(NINT) NINTP(1)=LTOTAL
DO 1400 K=KL, KK, KJUMP
DO 1400 N=1, NMAX
C.....EQ. 61 OR 62
DSKHN=(IRREGE(I,1,N,K)+IRREGE(I,2,N,K)*ABDETA)*SDETA
C.....EQ. 51
ALFA(N,K,1)=ALFA(N,K,1)+BJM*(SK(K)*HHN(N)-DSKHN)
1400 CONTINUE
SI 1420 IF (NOTI) GO TO 1440
DUM3=DETA2*ALOG(ABDETA)
DO 1430 P=PL, PU
X=DUM1+DUM2*CHICP(P)
CALL INFFUN
LTOTP(P)=LTOTAL+LTOTP(P)
IF(NINT) NINTP(P)=LTOTAL
DO 1430 K=KL, KK, KJUMP
DO 1430 N=1, NMAX
1430 ALFA(N,K,P)=ALFA(N,K,P) + BJM*SK(K)*(HHN(N)+IRREGI(N,P)*DUM3)
1440 IF (NOTT) GO TO 1460
X=DUM1+DUM2
CALL INFFUN
LTOTP(PP)=LTOTAL+LTOTP(PP)
IF(NINT) NINTP(PP)=LTOTAL
DUM4=ABDETA*SDETA
DO 1450 K=KL, KK, KJUMP
DO 1450 N=1, NMAX
C.....EQ. 71 OR 72
DSKHN=IRREGE(I,3,N,K)*DUM4
1450 ALFA(N,K,PP)=ALFA(N,K,PP)+BJM*(SK(K)*HHN(N)-DSKHN)

```

```

1460 CONTINUE
      IF(NINT) WRITE(U6,8) (NINTP(P),P=1,PP)
1600 CONTINUE
C..... ***** SPANWISE INTEGRATION FINISHED *****

```

```

C
C.....CHANGING TO MODES SK=SIN(K*THETA)
C
      IF(KK.EQ.1) GO TO 1609
      KL2=KL+2
      DO 1608 P=1,PP
      DO 1608 N=1,NMAX
      IF(MODES.NE.0) ALFA(N,2,P)=2.*ALFA(N,2,P)
      IF(KK.LT.3) GO TO 1608
      DO 1606 K=KL2,KK,KJUMP
1606 ALFA(N,K,P)=2.*ALFA(N,K,P)+ALFA(N,K-2,P)
1608 CONTINUE
1609 CONTINUE

```

```

C
52 C.....END OF CHANGING MODES
C
      IF(INFSTO.NE.0) GO TO 1610
      NROW2=PP*M
      NROW1=NROW2-PP+1

```

```

C
C.....THE INFLUENCE MATRIX IS STORED AS SINGLE PRECISION
C

```

```

      DO 1605 P=1,PP
      DO 1605 K=KL,KK,KJUMP
      DO 1605 N=1,NMAX
1605 ALFA2(N,K,P)=ALFA(N,K,P)
      WRITE(U11) (((ALFA2(N,K,P),N=1,NMAX),K=KL,KK,KJUMP),P=1,PP)
      WRITE(U6,5)NROW1,NROW2
1610 IF(IWRITE.EQ.0) GO TO 1650
      DO 1630 P=1,PP
      WRITE(U6,2) ETAMS,CHICP(P),LTOTP(P)
      DO 1630 K=KL,KK,KJUMP
      WRITE (U6,3) K,P
      WRITE(U6,4) (ALFA(N,K,P),N=1,NMAX)
1630 CONTINUE

```



```

1650 CONTINUE
DO 1160 P=1,PP
GTOTAL=GTOTAL+LTOTP(P)
1160 CONTINUE
2000 CONTINUE
WRITE(U6,6)GTOTAL
RETURN
1 FORMAT(22H1STARTING 2000 M LOOP //)
2 FORMAT(5H0ETA=, F10.7, 6H CHI=, F9.7, 13H #INT,PTS,= ,15)
3 FORMAT (9H (ALFA(N, ,12, 1H, , 12, 11H),N=1,NMAX))
4 FORMAT (1X,F14.7,3F15.7)
5 FORMAT(5HOROWS ,13,1H=, 12, 17H HAVE BEEN STORED )
6 FORMAT(20H0INT, POINT TOTAL = ,18)
7 FORMAT(4H0M = ,13)
8 FORMAT(1X, 26I5)

```

END

SUBROUTINE INFFUN

C.....THIS SUBROUTINE DETERMINES THE PRESSURE MODE INFLUENCE  
C.....FUNCTIONS, WHICH ARE THE INTEGRAL ON (-1,+1) OF

53 C..... HN(XP)\*KBAR(X=XP,Y)/2.  
C.....HN ARE THE VAN SPEIGEL CHORDWISE FUNCTIONS  
C.....THE INTEGRATION IS DONE REPEATEDLY WITH  
C.....TWICE AS MANY INTEGRATION POINTS EACH TIME  
C.....UNTIL EITHER THE MAXIMUM CHANGE IS LESS  
C.....THAN DELTA AND LL,GE,RLLMIN OR UNTIL LL=LLMAX1.  
C.....LMIN IS THE STARTING NUMBER OF INTEGRATION POINTS.  
C.....NESTING IS USED SO THAT AFTER THE FIRST INTEGRATION THE  
C.....VALUES OF KBAR NEED ONLY BE DETERMINED AT  
C.....EVERY OTHER STATION.  
C.....THE INTEGRAL IS COMPUTED USING THE QUADRATURE FORMULA  
C.....OF PROBLEM 37, P.364 OF HILDEBRAND'S NUMERICAL METHODS.  
C.....LLMAX =(LMIN+1)\*2\*\* (LPWR-1)-1  
DOUBLE PRECISION SUM,P1  
DIMENSION G(10,512),XP(512),HHN(10),HHNR(10),SUM(10),DIFFN(10)  
REAL KBAR  
INTEGER U6  
LOGICAL SUPRES,NOCCI  
COMMON P1

```

COMMON LLMAX,LPWR,LLMIN,LLMAX1,G,XP,X,YS,DELTA,HHN,NMAX,SUPRES,NOC
1CI,RLLMIN,UA,LTOTAL
  XMP=X+1.
  KBAR=(1.+XMP/SQRT(XMP*XMP+YS))/2.
  DO 10 N=1,NMAX
    SUM(N)=KBAR*G(N,LLMAX1)
10 HHN(N)=0.
20 CONTINUE
  LL=LLMIN+1
  LJUMP=LLMAX1/LL
  LJUMP2=LJUMP
C.....EACH PASS THROUGH THE J LOOP REPRESENTS A CHORDWISE
C.....INTEGRATION
  DO 200 J=1,LPWR
C.....PREVIOUS RESULTS ARE SAVED IN HHNR
  DO 30 N=1,NMAX
    HHNR(N)=HHN(N)
  30 CONTINUE
54 C.....LL= NUMBER OF INTEGRATION POINTS INCLUDING THE
C.....ONE AT THE LEADING EDGE.
  DO 150 L=LJUMP,LLMAX,LJUMP2
    XMP=X-XP(L)
    KBAR=1.+XMP/SQRT(XMP*XMP+YS)
    DO 150 N=1,NMAX
      SUM(N)=SUM(N)+KBAR*G(N,L)
150 CONTINUE
    RLL=LL
    LJUMP2=LJUMP
    LJUMP=LJUMP/2
    DIFF=0
    DO 170 N=1,NMAX
C.....THE FOLLOWING STATEMENT INSURES THAT ONLY HHN'S COMPUTED
C.....WITH A CERTAIN DEGREE OF PRECISION WILL BE USED IN THE
C.....COMPARISON CONVERGENCE TEST.
C.....HHN(N) WILL = HHNINF(N) UNTIL THE REQUIRED D.O.P. IS
C.....OBTAINED.
      IF(LL=N,LT.=1)GO TO 175
      HHN(N)=SUM(N)/RLL
      DIFFN(N)=ABS(HHNR(N)-HHN(N))

```

```

170 DIFF=AMAX1(DIFF,DIFFN(N))
175 LL=LL+LL
180 CONTINUE
    IF (DIFF .LT. DELTA .AND. RLL.GE.RLLMIN) GO TO 300
200 CONTINUE
    LTOTAL=LL/2
    IF(SUPRES)RETURN
    WRITE (U6,1)
    GO TO 305
300 CONTINUE
    LTOTAL=LL/2
    IF(NOCCI)RETURN
305 WRITE (U6,2) X,YS,DELTA,LTOTAL,(DIFFN(N),N=1,NMAX)
    RETURN
1 FORMAT(////' THE CHORDWISE INTEGRATION FAILED TO CONVERGE')
2 FORMAT('      X =',F15.6,'      YS =',F15.6,' DELTA =',F12.8,'
NUMBER OF INTEGRATION POINTS =',I5,' DIFF =',5F12.8/8X,5F12.8)
END
SUBROUTINE INITAL(CWTYPE,PP)
55 C.....THIS SUBROUTINE COMPUTES .G., THE VALUES OF THE
C.....CHORDWISE PRESSURE MODES EVALUATED
C.....AT POSITIONS GIVEN BY THE ARRAY (XP), WHICH IS ALSO COM-
C.....EACH MODE HAS BEEN MULTIPLIED BY  $\pi/2 \cdot \sin(\phi)$ .
C.....THIS VERSION IS FOR THE VAN SPEIGEL PRESSURE MODES.
C.....QUANTITIES ARE DETERMINED AT EACH  $\phi = -\cos(\phi)$ .
C..... $\phi$  TAKES ON THE VALUES  $\phi_1, \phi_2, \dots, \phi_{LMAX+DPHI}$ ,
C.....AND THE LAST VALUE IS  $\phi = 0$  (I.E.,  $\phi(LMAX+1) = -1.$ ).
C.....SIMILARLY  $G(N,L)$  IS THE VALUE OF G FOR  $\phi = L + \phi_1$ ,
C.....EXCEPT THAT  $G(N,LMAX+1)$  IS THE VALUE FOR  $\phi = 0$ .
C.....DIMENSION AS  $G(NMAX, LMAX+1), XP(LMAX+1)$ 
    DIMENSION G(10,512),XP(512),HHN(10)
    COMMON PI
    COMMON LMAX,LPWR,LMIN,LMAX1,G,XP,X,YS,DELTA,HHN,NMAX
    COMMON SUPRES,NOCCI,RLLMIN,U6,LTOTAL
    DOUBLE PRECISION PI,DPHI,PHI,CNL,CNM1L,CNM2L
    INTEGER TLMAX1,CWTYPE,PP,U6
    LOGICAL SUPRES,NOCCI
    DATA LMAXX /511/,LMAXR,NMAXR/0,0/
    IF (LMIN.GT.0)GO TO 4

```

```

    LMIN=1
    IF(CWTYPE)4,2,3
2   IF(PP.GT.2)LMIN=PP-2
    IF(MOD(PP,2).NE.0.AND.PP.GT.4)LMIN=(PP-3)/2
    GO TO 4
3   LMIN=PP+PP
4   LMAX1=LMIN+1
    LMAX=MINO(LMAX,LMAX1)
    IF(LMAX.EQ.0)LMAX=LMAX1
    LMAX=MAX0(LMIN,LMAX)
    DO 5 K=1,20
    LPWR=K
    TLMAX1=LMAX1+LMAX1
    IF(TLMAX1.GT.LMAX+1)GO TO 6
5   LMAX1=TLMAX1
6   LMAX=LMAX1-1
    IF(LMAX.EQ.LMAXR .AND. NMAX.LE.NMAXR)RETURN
    LMAXR=LMAX
    NMAXR=NMAX
    L2=LMAX1/2
    DPHI=PI/DFLOAT(LMAX1)

```

56  
C  
C.....COMPUTING THE ARRAY .XP.

```

    DO 10 K=1,L2
    L3=LMAX1-K
    XP(L3)=DCOS(DFLOAT(K)*DPHI)
    XP(K)=-XP(L3)
10 CONTINUE
    XP(L2)=0
    XP(LMAX1)=1

```

C.....COMPUTING THE ARRAY .G.. ALL THE NECESSARY  
C.....NUMBERS HAVE BEEN COMPUTED AND ARE CONTAINED IN .XP..  
C.....IT IS ONLY NECESSARY TO COMBINE THEM IN THE APPROPRIATE  
C.....MANNER.

```

    DO 60 L=1,LMAX
    CNM2L=1
    NLM1=L-1
    INDEX=L
    CNM1L=-XP(INDEX)
60

```

```

G(1,L)=1+CNM1L
IF(NMAX.EQ.1) GO TO 60
DO 50 N=2,NMAX
  NLM1=N*L-1
  INDEX=MOD(NLM1,LMAX1)+1
  CNL=XP(INDEX)
  IF(INDEX.EQ.MOD(NLM1,TLMAX1)+1) CNL=-CNL
  G(N,L)=(CNM2L-CNL)/2.D0
  CNM2L=CNM1L
  CNM1L=CNL
50 CONTINUE
60 CONTINUE
  G(1,LMAX1)=2
  IF(NMAX.EQ.1) GO TO 80
  DO 70 N=2,NMAX
    G(N,LMAX1)=0
70 CONTINUE
80 CONTINUE
  RETURN
  END
  SUBROUTINE INFZY(X,NMAX,HHN)

```

57

```

C
C.....THIS SUBROUTINE COMPUTES THE INFLUENCE FUNCTIONS FOR Y=0.
C.....THIS VERSION OF THE PROGRAM IS FOR THE VAN SPEIGEL MODES.
  DIMENSION HHN(NMAX)
  DOUBLE PRECISION PHI,DN,SN,SNM1,SNM2,TOPI,PI
  DATA TOPI / .6366197723D0/,PI/3.141592653589793 00/
  PHI=DARCCOS(-DBLE(X))
  SNM2=DSIN(PHI)
  HHN(1)=TOPI*(PHI+SNM2)
  IF(NMAX.LT.2) RETURN
  SNM1=DSIN(2.D0*PHI)/2.D0
  HHN(2)=(PHI-SNM1)/PI
  IF(NMAX.LT.3) RETURN
  DO 10 N=3,NMAX
    DN=N
    SN=DSIN(DN*PHI)/DN
    HHN(N)=(SNM2-SN)/PI
    SNM2=SNM1

```

SNM1=SN  
10 CONTINUE  
RETURN  
END

C FUNCTION CEL2(Z,A,B)  
Z=K\*\*2= PARAMETER OF THE ELLIPTIC INTEGRALS

GEO=1.-Z  
IF(GEO)1,2,6  
1 RETURN  
C SET RESULT VALUE = OVERFLOW

2 IF(B)3,5,4  
3 CEL2=-1.E38  
RETURN

4 CEL2=1.E38  
RETURN

5 CEL2=A  
RETURN

58 C COMPUTE INTEGRAL

6 GEO=SQRT(GEO)

ARI=1,

AA=A

AN=A+B

W=B

7 W=W+AA\*GEO

W=W+W

AA=AN

AARI=ARI

ARI=GEO\*ARI

AN=W/ARI+AN

C TEST OF ACCURACY

IF(AARI-GEO-1.E-4\*AARI)9,9,8

8 GEO=SQRT(GEO\*AARI)

GEO=GEO+GEO

GO TO 7

9 CEL2=.7853982 \*AN/ARI

RETURN

END